Homework 2
APPM/MATH 4650 Fall '20 Numerical Analysis

Due date: Friday, September 11, before 5 PM, via Gradescope. **Instructor**: Prof. Becker
Theme: Convergence rates and rootfinding

Instructions Collaboration with your fellow students is OK and in fact recommended, although direct copying is not allowed. The internet is allowed for basic tasks (e.g., looking up definitions on wikipedia) but it is not permissible to search for proofs or to *post* requests for help on forums such as `http://math.stackexchange.com/` or to look at solution manuals. Please write down the names of the students that you worked with.

An arbitrary subset of these questions will be graded.

Turn in a PDF (either scanned handwritten work, or typed, or a combination of both) to **Gradescope**, using the link to Gradescope from our Canvas page. Gradescope recommends a few apps for scanning from your phone; see the Gradescope HW submission guide.

We will primarily grade your written work, and computer source code is *not* necessary except for when we *explicitly* ask for it (and you can use any language you want). If not specifically requested as part of a problem, you may include it at the end of your homework if you wish (sometimes the graders might look at it, but not always; it will be a bit easier to give partial credit if you include your code).

Problem 1: Convergence Rates.

a) Misleading Taylor Series. To find the convergence rate of a function f, often we use Taylor series. For example, if we say $(1+x)^n = 1 + nx + O(x^2)$ as $x \to 0$, the guess of $O(x^2)$ was determined from the Taylor series $(1+x)^n = 1 + nx + n(n-1)x^2/2! + \ldots$. However, we need to be careful about the radius of convergence of the Taylor series and that the higher-order derivatives are bounded. Show that e^x is *not* $O(x)$ for $x \to \infty$.

b) Show that $x \sin \sqrt{x} = \theta(x^{3/2})$ as $x \to 0$.

c) Show that $e^{-t} = o(\frac{1}{t^2})$ as $t \to \infty$.

d) Show that $\int_0^\varepsilon e^{-x^2} dx = O(\varepsilon)$ as $\varepsilon \to 0$. *Hint*: one nice way to do this uses the Mean Value Theorem.

e) Show that $-x/\log(x)$ is $o(x)$ but not $O(x^2)$ as $x \to 0$.

Problem 2: Write a function `bisect` which calls the bisection method, and takes as in put a function `f`, numbers specifying the interval $[a, b]$, and a tolerance `tol` which controls how far the approximate root is from the true root.

Make sure to notify the user in some way (e.g., raising an error, or returning an informative exit code) is the given function does *not* change signs on $[a, b]$. Optionally, you can also specify a maximum number of iterations.

The deliverable for this problem is your **code** for this function `bisect`. You may use any programming language, though Python and Matlab are encouraged.

Note: In Matlab, the input function might be defined in another file, like in `myfun.m`. In this case, to call your bisection function, use it like `bisect(@myfun,...)`. The `@` tells Matlab to make a function handle. Alternatively, if you defined `myfun` as an anonymous function, then you don't need the `@`, e.g., `fcn=@(x)x^2; bisect(fcn, ...)`.

Problem 3: Consider the equation $2x - 1 = \sin x$.

a) Find a closed interval $[a, b]$ on which the equation has a root r, and use the Intermediate Value Theorem to prove that r exists.

b) Prove that r from (a) is the only root of the equation (on all of $\mathbb{R}$).

c) Use your function from Problem 2 to approximate r to eight correct decimal places. The **deliverable** is a list of the approximation at every step.

d) The function $f(x) = (x-5)^9$ has a root (with multiplicity 9) at $x = 5$ and is monotonically increasing (decreasing) for $x > 5$ ($x < 5$) and should thus be a suitable candidate for your function above. Use `a=4.82` and `b=5.2` and `tol = 1e-4` and use `bisection` with:

   i. $f(x) = (x-5)^9$.

   ii. The expanded expanded version of $(x-5)^9$, that is, $f(x) = x^9 - 45x^8 + \ldots - 1953125$. You may use `polyval` or `numpy.polyval`

   The **deliverables** for this problem are (1) a graph of the error produced from both variants discussed above, and (2) a discussion of what you think is happening.

Problem 4: Tent Map. Consider the "tent function"

$$g_\mu(x) = \mu \min(x, 1-x)$$

on the interval $[0, 1]$, where $0 \leq \mu \leq 2$. This is like an upside down and shifted absolute value function (suggestion: graph it, for both $\mu < 1$ and $\mu > 1$). We're interested in fixed points of the tent map, so solutions to $x = g_\mu(x)$. This is a favorite example used when teaching dynamical systems. *Suggestion:* Try making a "cobweb plot" of this function using, e.g., geogebra.org/m/uvsfvNDt. The software may not like functions defined with a "max" function, so you can instead use the fact that

$$\min(x, 1-x) = -\left|x - \frac{1}{2}\right| + \frac{1}{2}. \tag{1}$$

a) Show that for $\mu \in [0, 2]$ that $g_\mu(x) \in [0, 1]$ for all $x \in [0, 1]$.

b) Prove that for $0 \leq \mu < 1$ there is a unique fixed point in $[0, 1]$. Would the fixed point iteration find this fixed point? Does the contraction mapping theorem apply? *Hint:* The triangle inequality or reverse triangle inequality may be useful[1]

c) What are the fixed points if $\mu = 1$?

d) Prove that for $1 < \mu \leq 2$ there are two fixed points in $[0, 1]$, and find these fixed points. Would the fixed point iteration find either of these fixed points? Does the contraction mapping theorem apply?

e) Run the fixed point iteration starting at $x = \pi/6$ for both $\mu = 1.1$ and $\mu = 1.5$. Do this for about 10 iterations; do you get a rough feel for what is happening? Then run 10^6 iterations and make a histogram (still for both values of μ). Include the histogram plots with your homework.

Tips for exporting jupyter notebook code to a PDF: try `nbconvert` which requires `pandoc`. You can do this on Colab, following the instructions here (but note that you may need to add a backslash before any white space when you run commands, e.g., change a command like
`!cp drive/My Drive/Colab Notebooks/Untitled.ipynb ./` to
`!cp drive/My\ Drive/Colab\ Notebooks/Untitled.ipynb ./`).

Note that if you include latex in the jupyter notebook, when you run `nbconvert`, you cannot have any whitespace near the \$ symbols for math due to a requirement of `pandoc` (see here). So, `$ f(x) = 3x^2 $` will not work, but `$f(x) = 3x^2$` will be OK.

[1] The triangle inequality says that you cannot get from point a to point b any faster than taking a direct path. Mathematically, it is $|a-b| \leq |a| + |b|$ (as well as $|a+b| \leq |a| + |b|$). In one dimension, you can prove this by considering all the possible cases for the sign of a and b. In higher dimensions, the triangle inequality is always true by definition for any *norm*, since norms are required to satisfy the triangle inequality. The *reverse triangle inequality* is just the triangle inequality with a $a = a + 0$ trick, i.e., $|a| = |a+0| = |a-b+b| \leq |a-b| + |b|$ so rearranging gives $\boxed{|a-b| \geq |a| - |b|}$.